

```
In [1]: %matplotlib notebook
import math
import os
import numpy as np
import sympy as sym
import pandas as pd
from matplotlib import rc
import matplotlib.pyplot as plt
from IPython.display import display, Math, Markdown
rc('font', **{'family': 'serif', 'serif': ['Computer Modern'], 'size': 20})
rc('text', usetex=True)
path = "C:/Users/Thomas/Laboratorio_Intermedio/Franck-Hertz/Data/"
path_ = "C:/Users/Thomas/Laboratorio_Intermedio/Franck-Hertz/Graphs/"
plt.rcParams["savefig.directory"] = path_
```

```
In [2]: def ChangeType(array):
    array_ = array.copy()
    for i in range(0, len(array)):
        array_[i] = float(array[i].replace(",", "."))
    return array_

def DataAnalysis(directory, index):
    """
    Esta función tiene como trabajo el tomar los datos de cada carpeta y someterlos al análisis estadístico
    requerido para las mediciones del experimento de Franck-Hertz. Cada dato tendrá un error asociado a la desviación
    estándar asociada a los puntos de ese mismo valor de voltaje.
    """

    Base = []
    for i in range(0, len(index)):
        base = pd.read_csv(path + directory + index[i], sep = " ")
        Base.append(base)

    BaseCurrents = []
    BaseVoltages = None
    limit = None

    for i in range(0, len(index)):
        # Solo se almacena un conjunto de voltajes (son los mismos para todas las mediciones).
        if i == 0:
            voltage = Base[i]["U1/V"].to_numpy()
            voltage = ChangeType(voltage)
            voltage = voltage[voltage <= 50]
            BaseVoltages = voltage
            limit = len(voltage)

        # Se toman los valores de corriente asociados al límite establecido.
        # (Sólo los asociados a voltajes menores o iguales a 50 V)
        current = Base[i]["IA/nA"].to_numpy()
        current = ChangeType(current)[0:limit]
        BaseCurrents.append(current)

    # A partir de esto se trabaja con el promedio de los datos en cada punto, para luego determinar
    MeanBase = np.array([])
    ErrorBase = np.array([])

    for i in range(0, len(BaseVoltages)):
        simpleSet = np.array([])
        for j in range(0, len(index)):
            simpleSet = np.append(simpleSet, BaseCurrents[j][i])

        value = np.mean(simpleSet)
        error = np.std(simpleSet)

        MeanBase = np.append(MeanBase, value)
        ErrorBase = np.append(ErrorBase, error)

    return (BaseVoltages, MeanBase, ErrorBase)
```

Bitácora - Experimento de Franck-Hertz

Hecho por: Simón Felipe Jiménez Botero & Thomas Andrade Hernández.

En el presente documento se encuentran desglosados los resultados de las mediciones asociadas al montaje experimental de Franck-Hertz. A lo largo de este código se podrán observar los resultados de cada medición, sumado a una breve explicación previa acerca de aquello que estamos haciendo, cómo lo estamos haciendo y alguna que otra cosa a tener presente para su replicación.

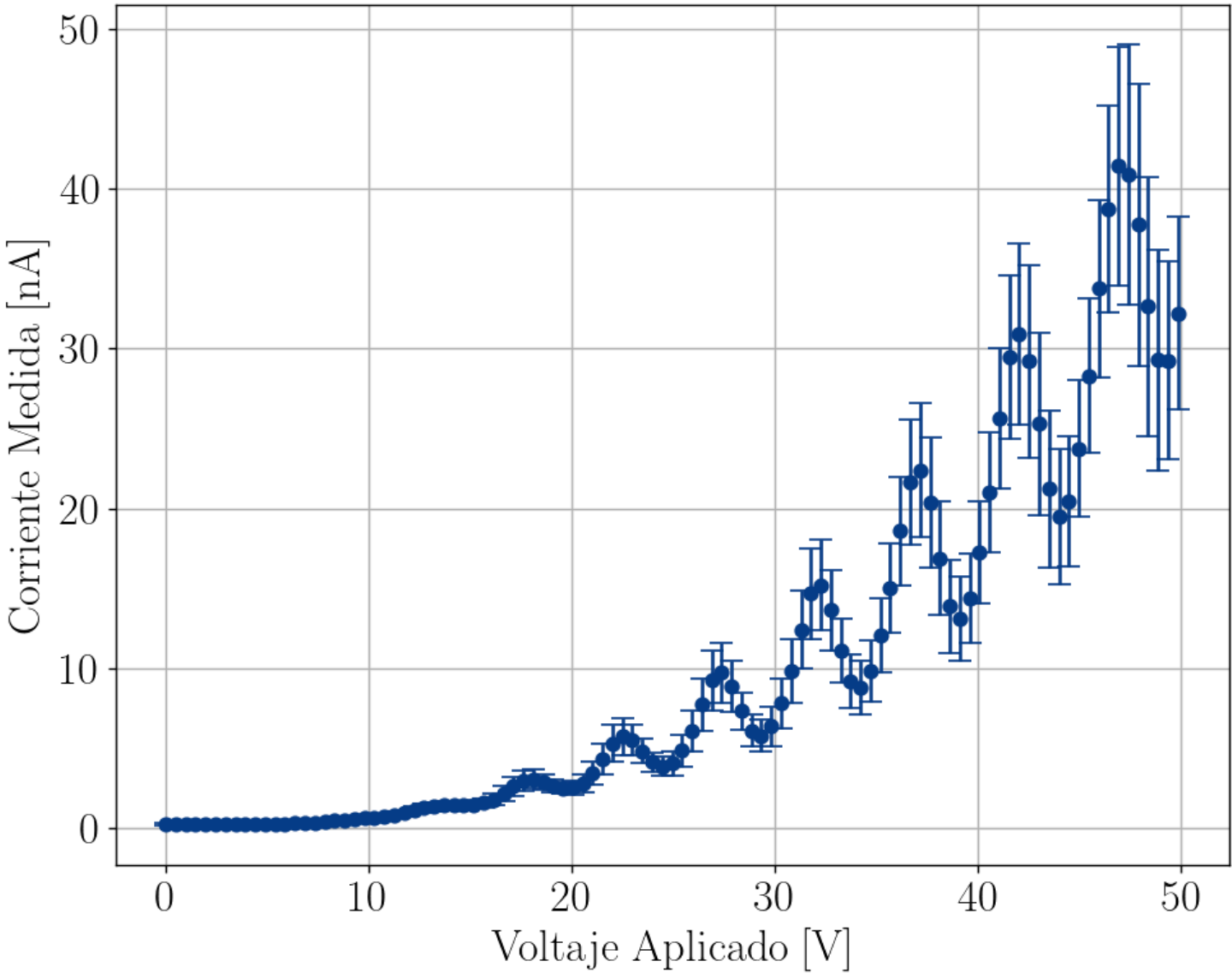
Actividad #1: Análisis del comportamiento de los datos.

En esta parte inicial tomamos múltiples mediciones para un mismo valor de temperatura (180°). Partiendo del comportamiento de las gráficas obtenidas, se decidió tomar las cinco mediciones realizadas y realizar un estudio estadístico que consiste en que, para un mismo valor de voltaje, tomar los distintos valores de corriente detectados y, posteriormente, realizar un promedio y una desviación estándar de los datos para así determinar el error.

```
In [3]: index = ["I", "II", "III", "IV", "V"]
        directory = "Base/Base "

In [4]: BaseVoltages, MeanBase, ErrorBase = DataAnalysis(directory, index)

In [5]: N = 20
        plt.figure(figsize = (9, 7))
        plt.errorbar(BaseVoltages[:,N], MeanBase[:,N], yerr = ErrorBase[:,N], color = "#083c87", capsize = 5, fmt = "o")
        plt.grid(True)
        plt.ylabel("Corriente Medida [nA]")
        plt.xlabel("Voltaje Aplicado [V]")
```



```
Out[5]: Text(0.5, 0, 'Voltaje Aplicado [V]')
```

Como se logra apreciar, las mediciones coinciden mucho para valores de voltaje por debajo de los 20 V, mientras que, al elevar este valor, la dispersi3n de los datos se hace mucho m1s evidente. Por este motivo, para las actividades posteriores se decidi3 tomar varias mediciones para un mismo valor de temperatura para as3 observar el comportamiento "real" de los datos.

Actividad #2: Comportamiento asociado al cambio t3rmico:

En la presente sesi3n se mostrar1 los resultados obtenidos de las tendencias para determinados valores de temperatura. Cabe aclarar que, debido al poco control por sobre la temperatura que pudimos tener, las oscilaciones t3rmicas en las mediciones fueron considerablemente altas. Se intent3 minimizarlas pero es necesario reconocer que, para cada valor de temperatura hay asociado un error aproximado de 2° por izquierda y derecha.

```
In [6]: Temperatures = ["Activity I/Heat 185-180/", "Activity I/Heat 190-185/", "Activity I/Heat 195-190/", "Activity I/Heat 200-185/"]
        TempNames = ["185°-180°", "190°-185°", "195°-190°", "200°-195°", "206°-201°"]
        TempIndexes = [{"I", "II", "III"}, {"I", "II", "III", "IV"}, {"I", "II", "III", "IV"}, {"I", "II", "III", "IV"}, {"I", "II", "III", "IV"}]

In [7]: TermicData = []

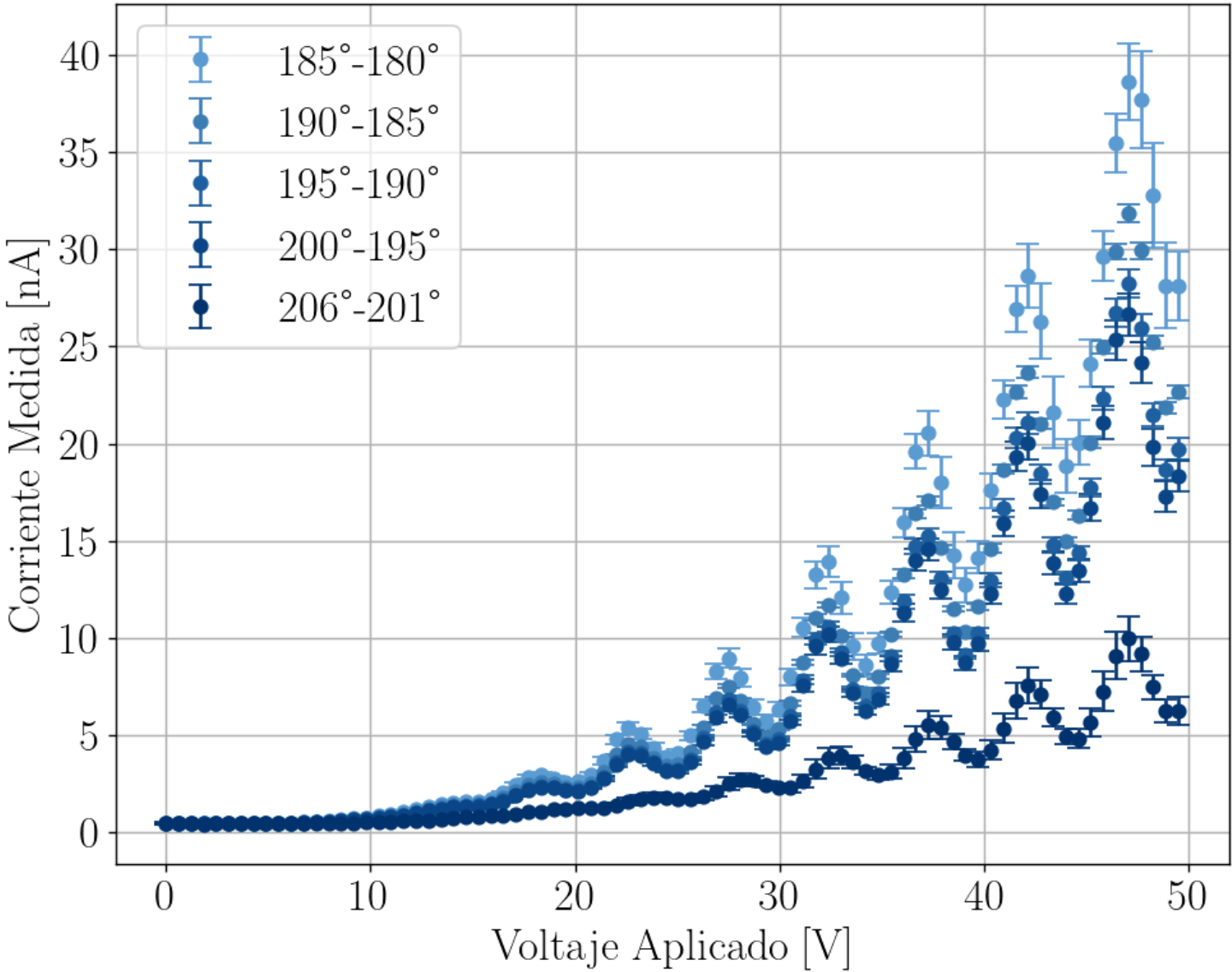
        for i in range(0, len(Temperatures)):
            Voltages, Mean, Error = DataAnalysis(Temperatures[i], TempIndexes[i])
            TermicData.append([Voltages, Mean, Error])

In [8]: plt.figure(figsize = (9, 7))
        Colors = ["#5A9BD6", "#3C7FB8", "#1F63A0", "#0A4A87", "#003470", "#001D59"]

        N = 25
        for i in range(0, len(TermicData)):
            plt.errorbar(TermicData[i][0][:N], TermicData[i][1][:N], yerr = TermicData[i][2][:N], color = Colors[i], label = TempNames[i])

        plt.ylabel("Corriente Medida [nA]")
        plt.xlabel("Voltaje Aplicado [V]")
```

```
plt.legend()  
plt.grid(True)
```

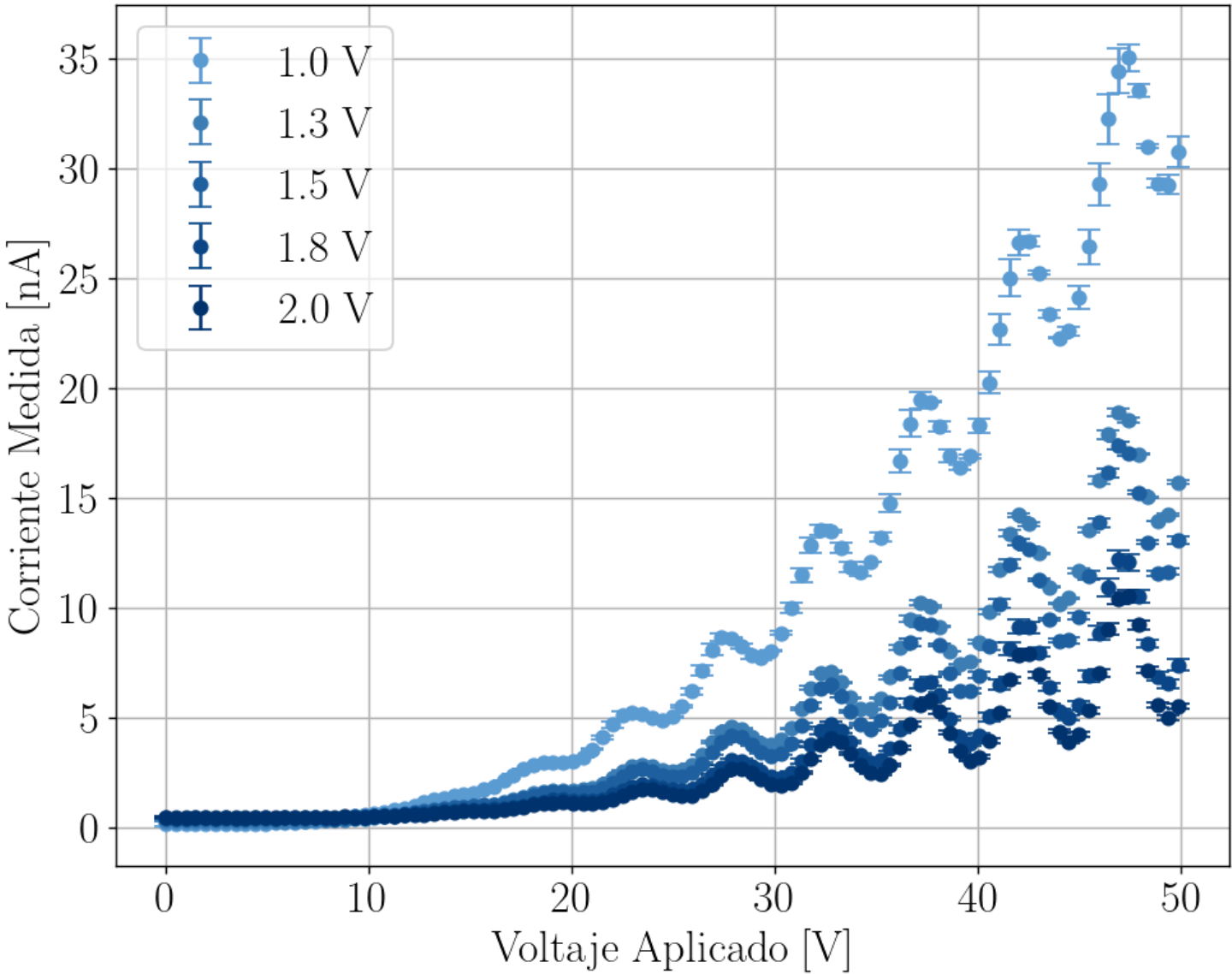


Se logra apreciar que, a mayor temperatura, menor es la corriente medida. La explicación a este fenómeno aún no la tenemos bien definida, por lo que se reservará el comentario para el informe.

Actividad #3: Comportamiento asociado al cambio de UA:

En esta sesión se mostrará los resultados obtenidos de las tendencias para diferentes valores de U_A .

```
In [9]: UAs = ["Activity II (200-205)/UA 1.0/", "Activity II (200-205)/UA 1.3/", "Activity II (200-205)/UA 1.5/", "Activity II  
UANames = ["1.0 V", "1.3 V", "1.5 V", "1.8 V", "2.0 V"]  
UAIndexes = [["I", "II"], ["II", "III"], ["I", "II", "III"], ["I", "II", "III"], ["I", "II", "III"]]  
  
In [10]: UAData = []  
  
for i in range(0, len(UAs)):  
    Voltages, Mean, Error = DataAnalysis(UAs[i], UAIndexes[i])  
    UAData.append([Voltages, Mean, Error])  
  
In [11]: plt.figure(figsize = (9, 7))  
Colors = ["#5A9BD6", "#3C7FB8", "#1F63A0", "#0A4A87", "#003470", "#001D59"]  
  
N = 20  
for i in range(0, len(UAData)):  
    plt.errorbar(UAData[i][0][:N], UAData[i][1][:N], yerr = UAData[i][2][:N], color = Colors[i], label = UANames[i],  
  
plt.ylabel("Corriente Medida [nA]")  
plt.xlabel("Voltaje Aplicado [V]")  
plt.legend()  
plt.grid(True)
```



Se logra apreciar que, a mayores valores de U_A , menor es la corriente medida. La explicación de este comportamiento se deja para el informe.

Actividad #4: Comportamiento asociado al cambio de UH:

En esta sesión se mostrará los resultados obtenidos de las tendencias para diferentes valores de U_H .

```
In [12]: UHs = ["Activity III (200-205)/UH 5.8/", "Activity III (200-205)/UH 6.1/", "Activity III (200-205)/UH 6.6/", "Activity III (200-205)/UH 6.8/"]
UHNames = ["5.8 V", "6.1 V", "6.6 V", "6.8 V"]
UHIndexes = [["I", "II", "III", "IV"], ["I", "II", "III", "IV"], ["I", "II", "III", "IV"], ["I", "II", "III", "IV"]]

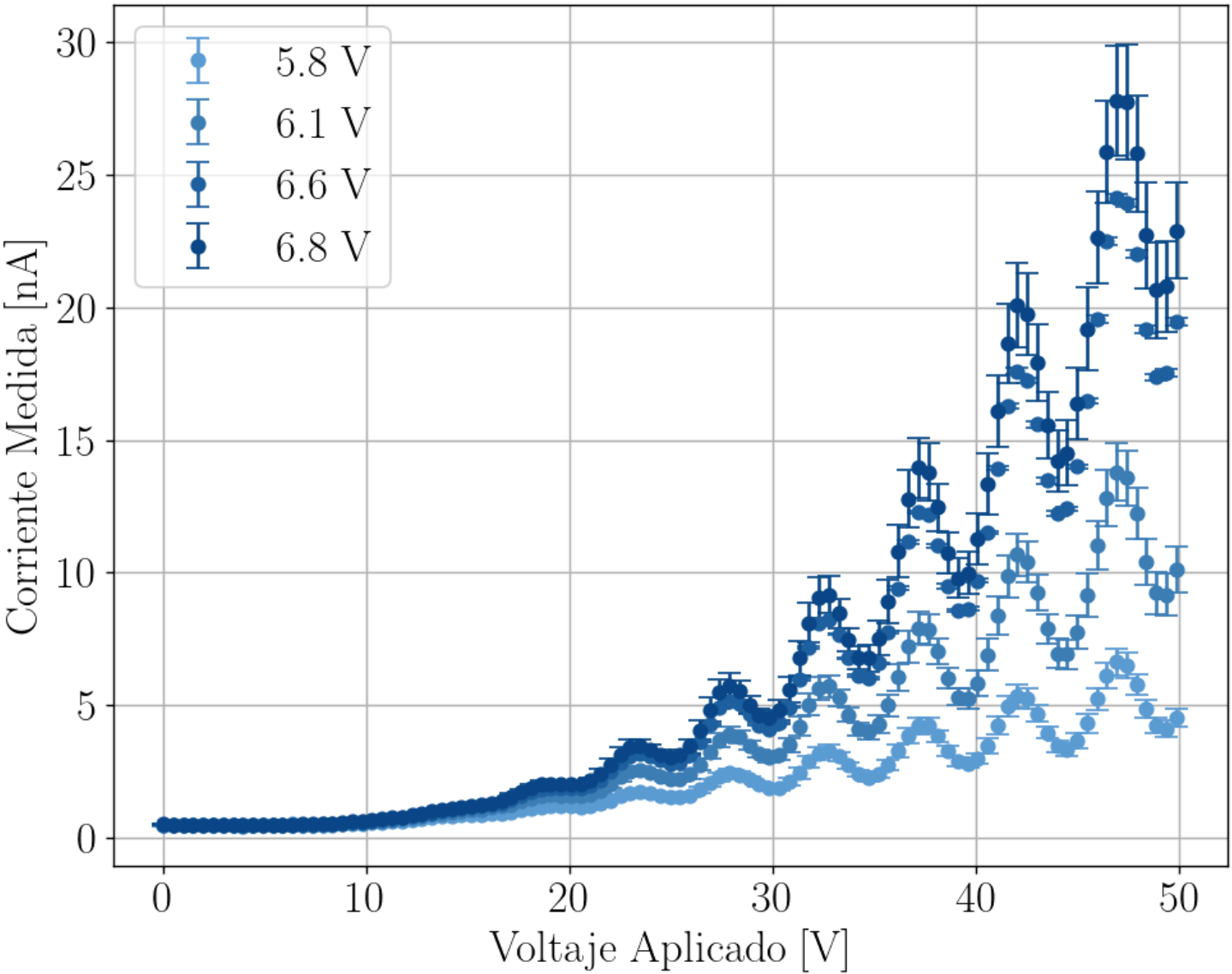
In [13]: UHData = []

for i in range(0, len(UHs)):
    Voltages, Mean, Error = DataAnalysis(UHs[i], UHIndexes[i])
    UHData.append([Voltages, Mean, Error])

In [14]: plt.figure(figsize = (9, 7))
Colors = ["#5A9BD6", "#3C7FB8", "#1F63A0", "#0A4A87", "#003470", "#001D59"]

N = 20
for i in range(0, len(UHData)):
    plt.errorbar(UHData[i][0][:N], UHData[i][1][:N], yerr = UHData[i][2][:N], color = Colors[i], label = UHNames[i],

plt.ylabel("Corriente Medida [nA]")
plt.xlabel("Voltaje Aplicado [V]")
plt.legend()
plt.grid(True)
```



Se logra apreciar que, a mayores valores de U_H , mayor es la corriente medida. Una vez más, la explicación de este comportamiento se deja para el informe :3.

La distancia entre mínimos ha de ser determinada al usar un filtro Gaussiano para suavizar la curva y poder estimar sus ubicaciones. Eso se hará posteriormente <3.